KPCI-1818　8 通道同步数据采集卡
使用说明书

北京科瑞兴业科技有限公司

北京科瑞兴业科技有限公司　　地址：北京市海淀区知春里 28 号开源商务写字楼 212、213 室
邮政编码：100086　　电话：010-51650651　010-62527214　　传真：010-62657424
http://www.krxgk.com
Sales E-mail: sgq@krxgk.com　　Tech Support E-mail: lilanzhen007@126.com

目录

第一章　　概述

1、　介绍
2、　应用
3、　性能和技术指标
4、　软件支持

第二章　　主要元件位置图、信号输出插座和开关跳线选择定义

1、主要元件布局图
2、短路套设置
3、信号输入输出插座定义
4、模拟信号输入连接方式及应注意的问题

第三章　函数模块调用说明

1．A/D 采集过程流程图
2．函数说明

第四章　KPCI-1818 8 通道同步数据采集卡的校准、保修和注意事项

1.　注意事项
2.　校准
3.　保修

KPCI-1818　8 通道同步 500KS/S 数据采集卡

第一章　概述

一、介绍：

KPCI-1818 是一款 PCI 总线 8 通道同步高速模拟量采集卡。使用 8 组同样的电路和 AD 芯片实现各通道的同步采集，当输入信号之间的时间关系很重要的时候，KPCI-1818 的这个特点可以满足用户要求。KPCI-1818 卡的最高采集频率可达 500KS/s，卡上带有 8K 容量 FIFO 存储芯片，可以完成大量信号的采集和存储，定时触发、软件触发和外触发三种方式，以适应不同场合的数据采集，方便了用户的使用。

KPCI-1818 卡上的 8 路开关量输入通道的输入电平与 TTL 兼容，6 路开关量输出通道的输出电平与 TTL 兼容。

KPCI-1818 卡上带有一个 4 位拨码开关，当计算机上安装多块板卡时，可使用此开关来定义每块板卡的 ID。用户可以很方便的在硬件配置和软件编程时区分和访问每块板卡。

- ◆ **总线：**32 位 PCI 总线，支持 PCI2.2 协议，即插即用。
- ◆ **PCI 芯片：**FPGA 接口芯片设计。
- ◆ **转换速度和精度：**12 位 A/D 转换器，通过率为 500K。
- ◆ **AD 通道数：**8 通道同步采集，差分输入。
- ◆ **转换方式：**支持软件查询方式、中断方式，外触发方式。
- ◆ **缓存：8**K 深度的 FIFO 缓存。
- ◆ **开关量：**8 路开关量输入通道输入电平与 TTL 兼容、6 路输出开关量输出通道电平与 TTL 兼容
- ◆ **软件支持：**提供动态链接库供编程使用，提供 VB，VC 编程示例。

二、性能及技术指标

2.1 PCI 局部总线性能

PCI 总线宽度 32 位，同步工作频率可达到 33MHz，最高传输速率为 132MB/S
能够完成自动配置，实现设备的即插即用

2.2 模拟信号输入部分

模拟通道数：8 路差分输入，同步采集
模拟输入电压范围：±5V, 0~10V。
模拟输入阻抗:　10MΩ

2.3 A/D 转换电路部分

A/D 分辨率：12 位(4096)
非线性误差：±1LSB(最大)
采集频率：500KS/S
系统测量精度：0.1%

2.4 A/D 采集方式及周期

三种采样触发方式：定时触发、软件触发和外触发，可由软件设置使用 FIFO 还是不使用 FIFO。

2.5 FIFO 存储器

深度： 8K Words

宽度： 12 位

标志： 空、有数、半满、全满

2.6 数字量 DIO：

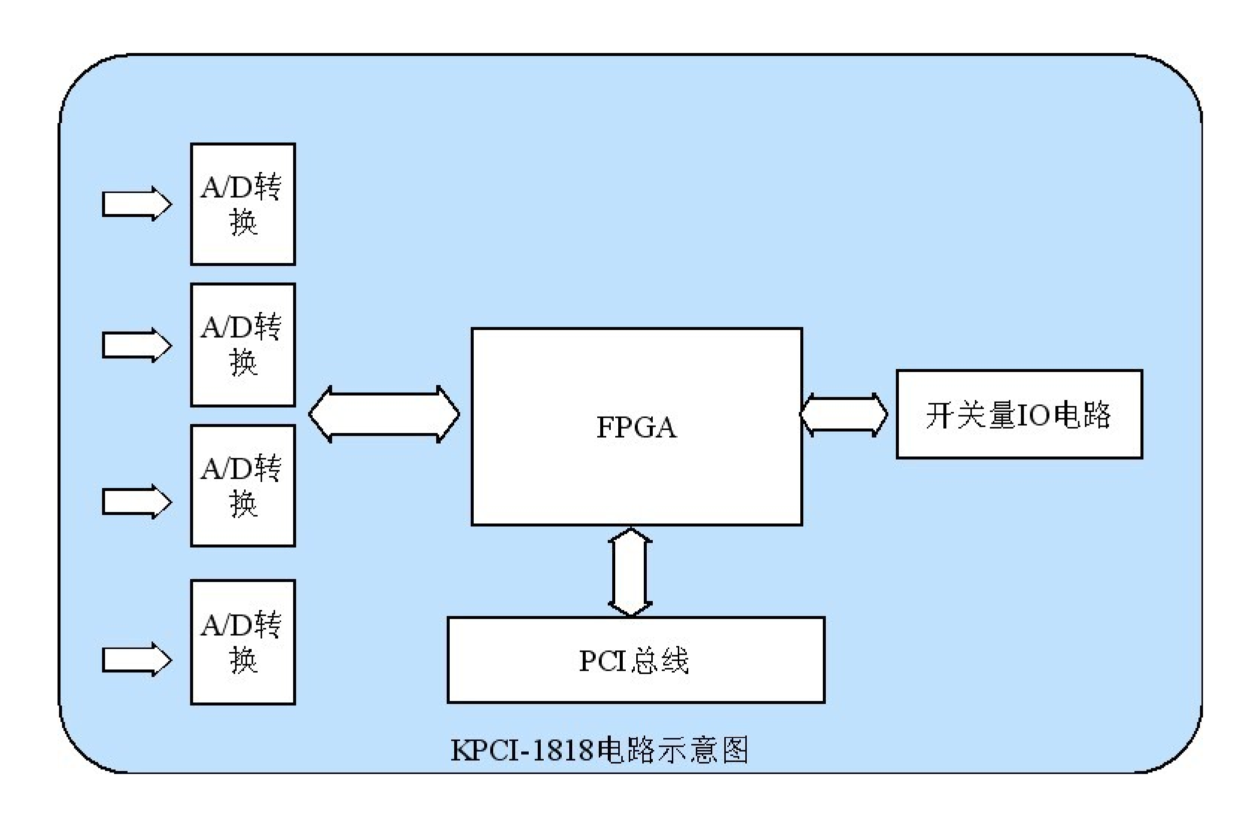
8 路数字量 DI 输入和 6 路数字量 DO 输出，兼容 TTL 电平。

三、软件支持

提供 Windows95/98/NT 下的多种语言的驱动，并提供在 VB 和 VC 环境下，开发的示例程序（详见软件说明部分）。

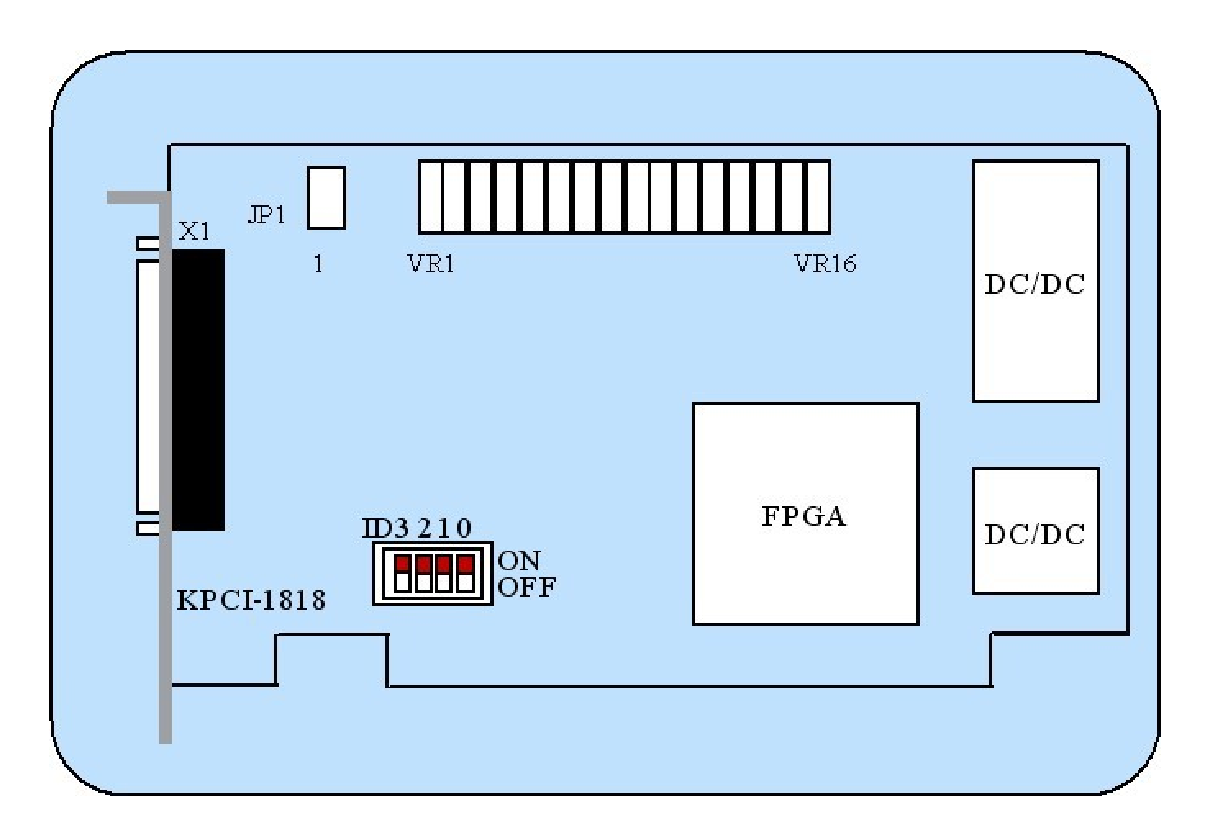
第二章 元件位置图、信号输出插座和开关跳线选择定义

一， 系统框图



KPCI-1818电路示意图

二、跳线和调整电位器位置说明



JP1 为模拟量输入信号极性选择，控制所有通道，每个通道不能单独控制。

选择 0~10V 时（为单极性），短接 2、3 脚。

选择±5V 时（为双极性），短接 1、2 脚。

VR1 为 A/D 第 1 通道的满度调整电位器。
VR2 为 A/D 第 1 通道的零点调整电位器。
VR3 为 A/D 第 2 通道的满度调整电位器。
VR4 为 A/D 第 2 通道的零点调整电位器。
VR5 为 A/D 第 3 通道的满度调整电位器。
VR6 为 A/D 第 3 通道的零点调整电位器。
VR7 为 A/D 第 4 通道的满度调整电位器。
VR8 为 A/D 第 4 通道的零点调整电位器。
VR9 为 A/D 第 5 通道的满度调整电位器。
VR10 为 A/D 第 5 通道的零点调整电位器。
VR11 为 A/D 第 6 通道的满度调整电位器。
VR12 为 A/D 第 6 通道的零点调整电位器。
VR13 为 A/D 第 7 通道的满度调整电位器。
VR14 为 A/D 第 7 通道的零点调整电位器。
VR15 为 A/D 第 8 通道的满度调整电位器。
VR16 为 A/D 第 8 通道的零点调整电位器。

三、信号输入插座定义

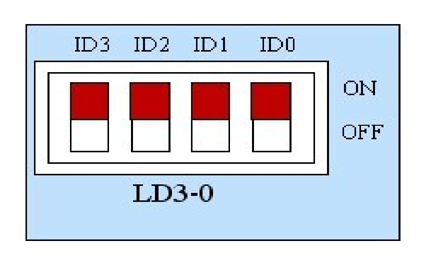
3.1 模拟量输入接口信号定义（输入插座为 DB37 孔式弯针插座）

D 型头管脚号	信号定义	D 型头管脚号	信号定义
1	Ain1+	20	Ain1-
2	Ain2+	21	Ain2-
3	Ain3+	22	Ain3-
4	Ain4+	23	Ain4-
5	Ain5+	24	Ain5-
6	Ain6+	25	Ain6-
7	Ain7+	26	Ain7-
8	Ain8+	27	Ain8-
9	AGND	28	AGND
10	AGND	29	Din1
11	Din2	30	Din3
12	Din4	31	Din5
13	Din6	32	Din7
14	Dout1	33	Dout2
15	Dout3	34	Dout4
16	Dout5	35	Dout6
17	GND	36	GND
18	EI	37	ES
19	OCLK	空	无

Ain1+～Ain8+ ： 双端模拟信号输入正端
Ain1- ～ Ain8- ： 双端模拟信号输入负端
AGND：模拟地
Din1 ～ Din8 ： 开关量输入通道
Dout1 ～ Dout16 ：开关量输出通道
GND：数字地
OCLK：时钟脉冲输出管脚，当时钟脉冲使能为使能时，输出脉冲信号最大为 500K
EI：外触发信号输入管脚，当外触发使能时此管脚接收到一个脉冲就转换一次
ES：开始采集脉冲输入管脚，当开始采集脉冲使能时，此管脚变成高电平后板卡开始采集数据

程序举例见软件说明书相应部分。

四、用四位拨码开关 LD3-0 可用来设置板卡标识 ID，当用户在同一计算机中使用多块 KPCI-1818 构建自己的系统时，ID 设置功能极为有用。用户可方便的在硬件配置和软件编程过程中区分和访问每块采集卡。



ID3	ID2	ID1	ID0	Board ID
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

注意： ON 为 0 OFF 为 1

五、模拟输入信号的连接方式

KPCI-1818板可按图5.1连接成模拟电压差分输入方式，可以有效抑制共模干扰信号，提高采集精度。8通道模拟输入信号正端接到Ain1+～Ain8+端，其模拟输入信号负端接到Ain1-～Ain8- 端， (见图5.1)

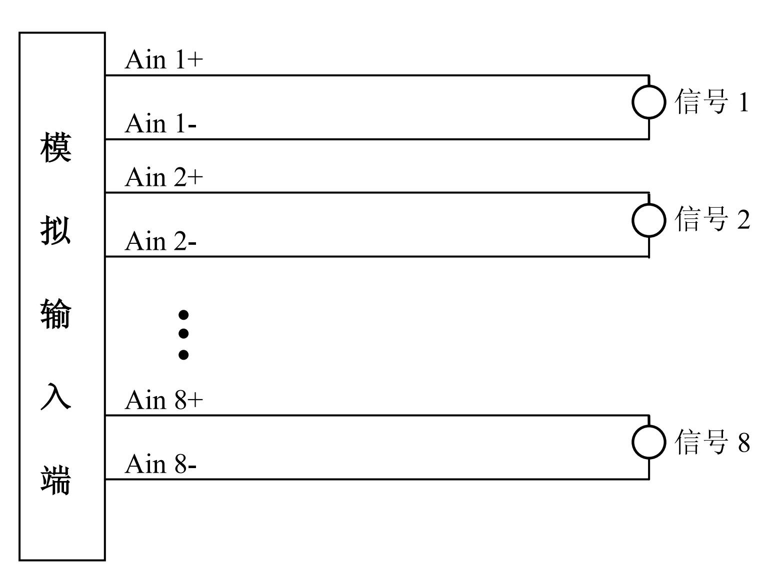


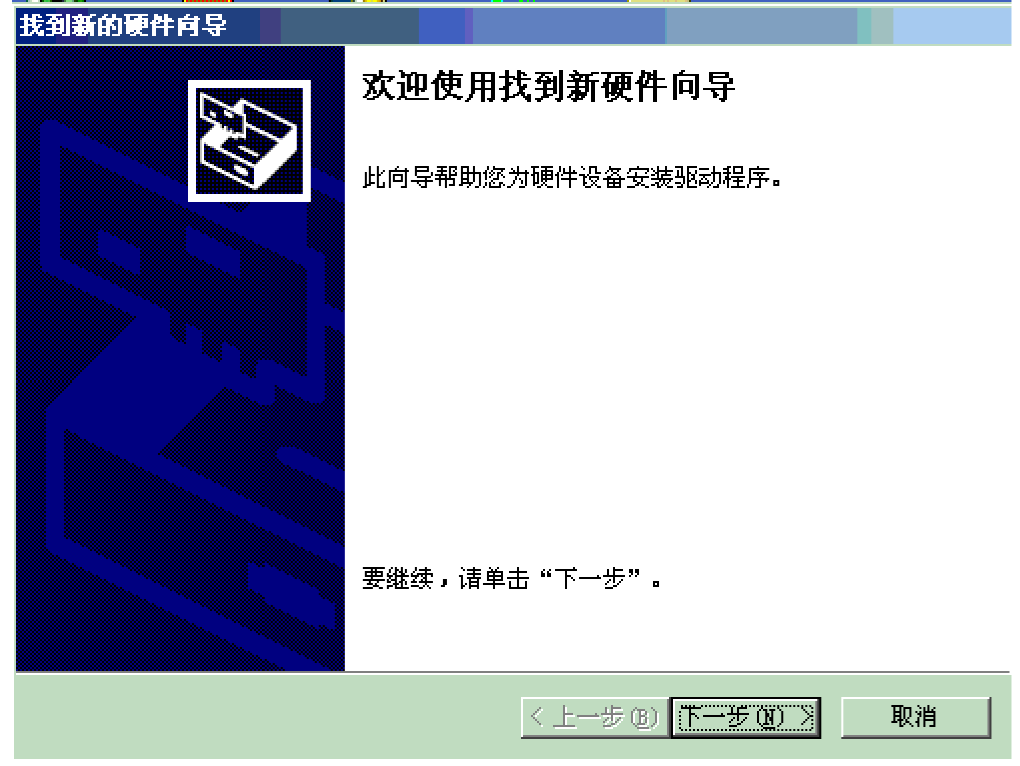
图 5.1 模拟量输入接线图

第三章 KPCI-1818 设备驱动程序安装

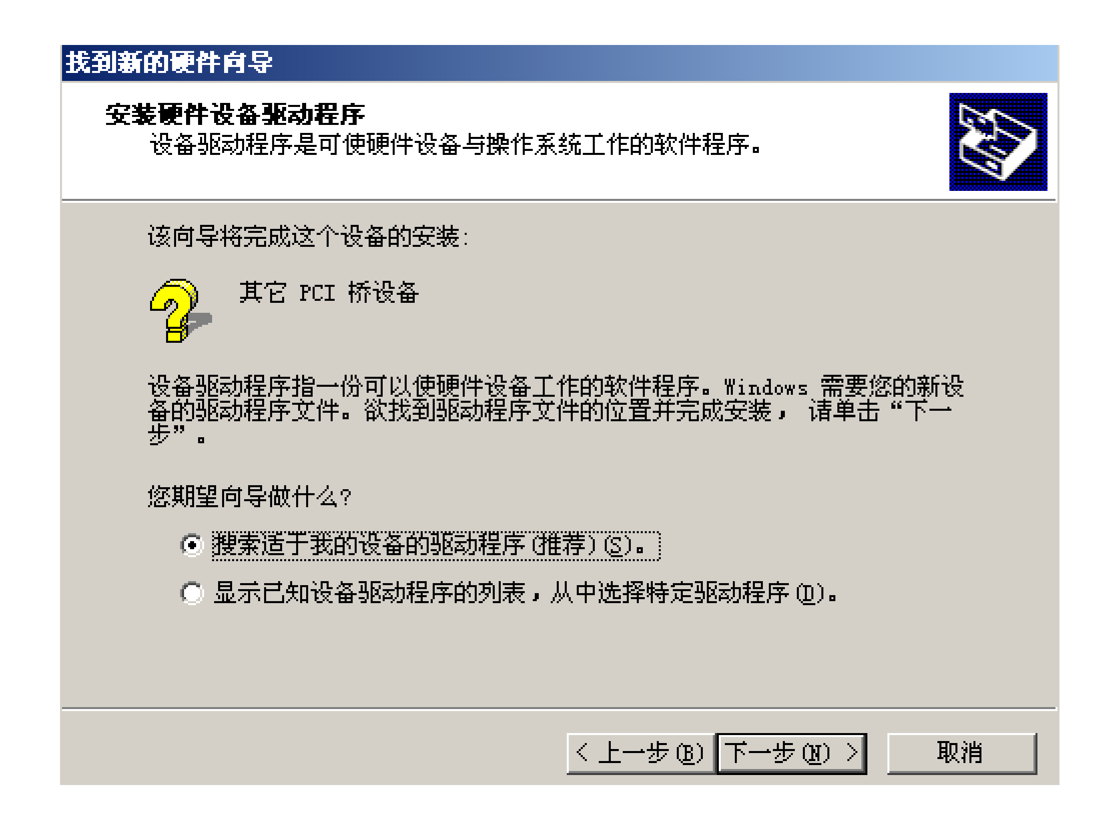
一、安装步骤

第一步 将 KPCI-1818 卡按硬件要求插入计算机主板上的任意一个 PCI 插槽中，并将其固定好，连接好其外接设备后，打开计算机电源，启动 Windows2000(XP)系统。

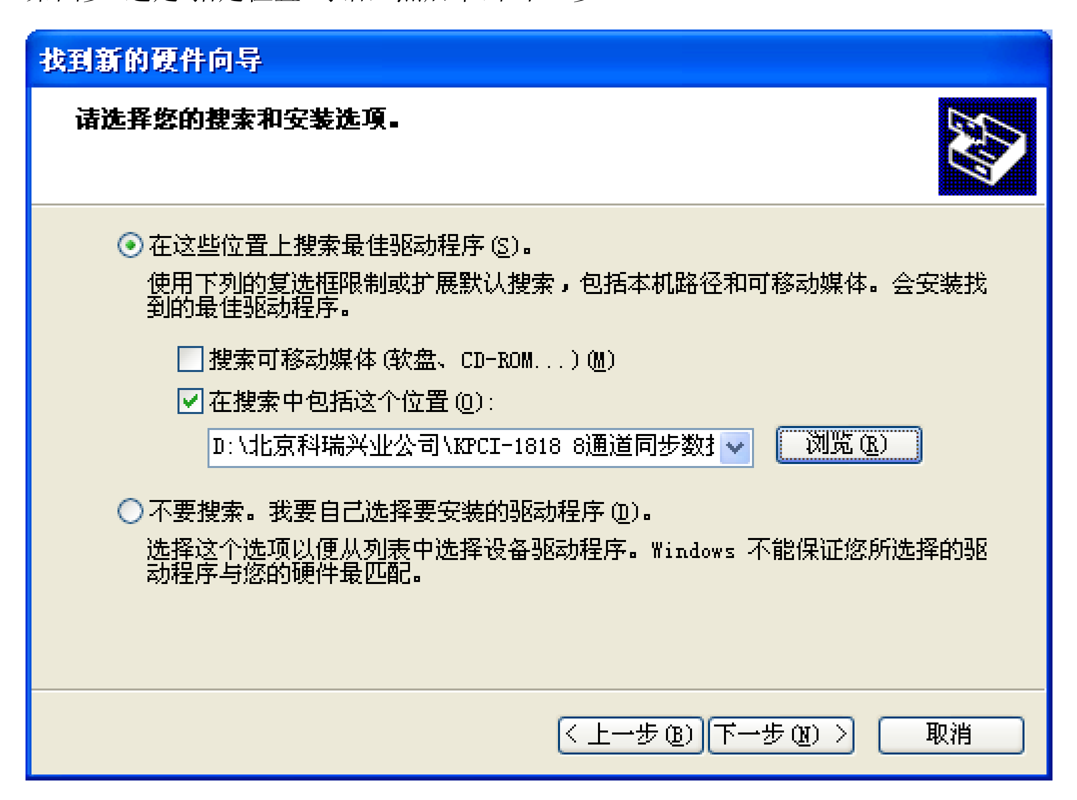
第二步 如果您正确地插好了 PCI 设备，Windows 系统在启动过程中便会发现这个新的 PCI 设备，并弹出 [欢迎使用找到新的硬件向导] 对话框，单击 [下一步] 按钮



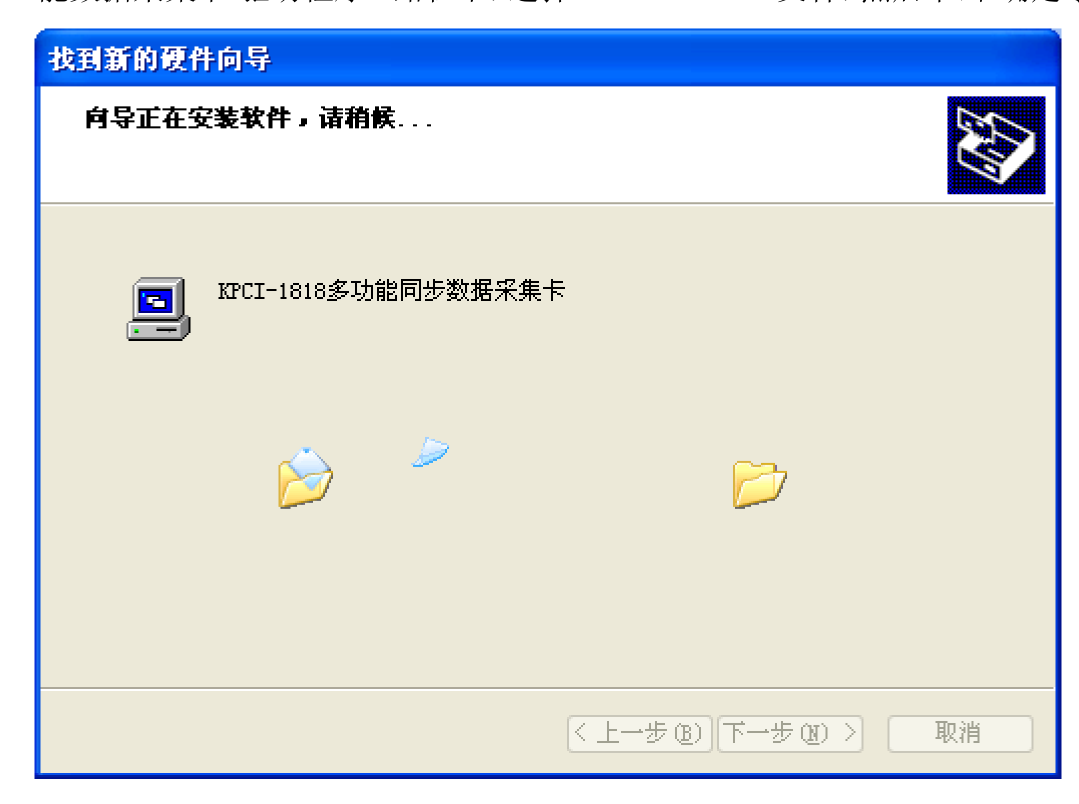
第三步 弹出[安装硬件设备驱动程序]对话框，在对话框中单击 [搜索适用于我的设备的驱动程序(推荐)(S)] 单选框，然后单击 [下一步] 按钮



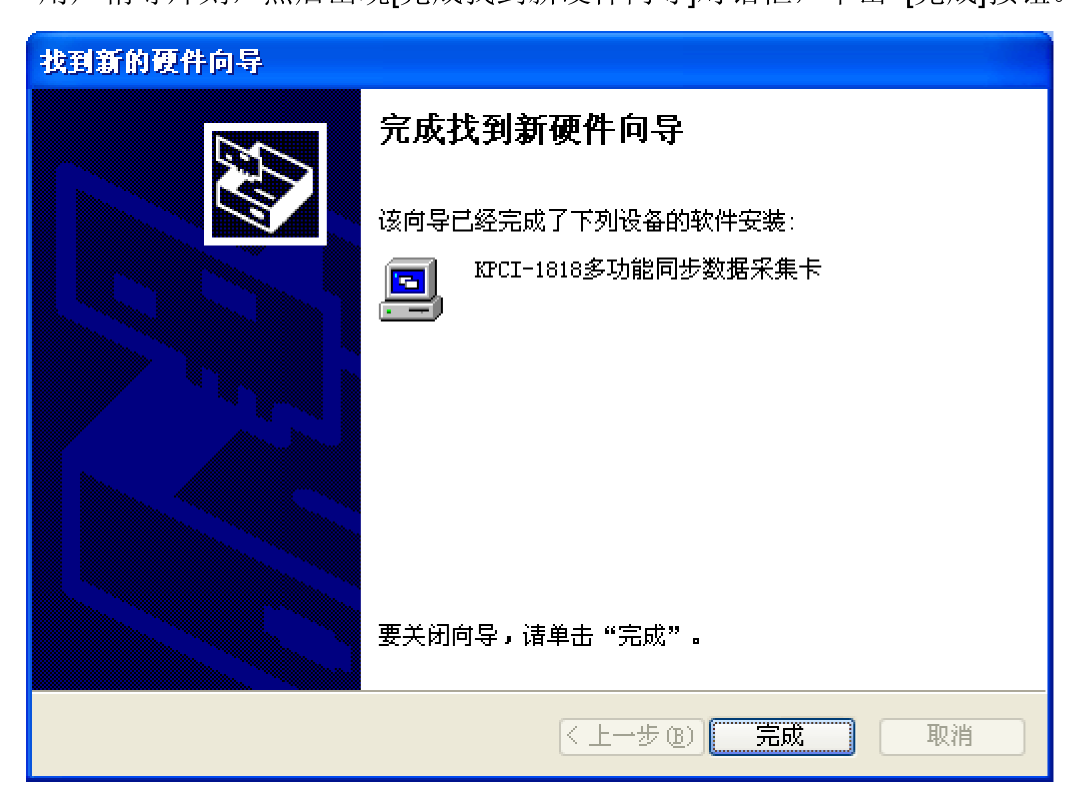
第四步 选定“指定位置”对话，然后单击“下一步”。



第五步 然后单击"浏览(R)"，将路径定位在光盘上的"PCI 总线测控板卡\KPCI-1811 多功能数据采集卡\驱动程序" 路径下，选择 KPCI1818.inf 文件，然后单击"确定"按钮。

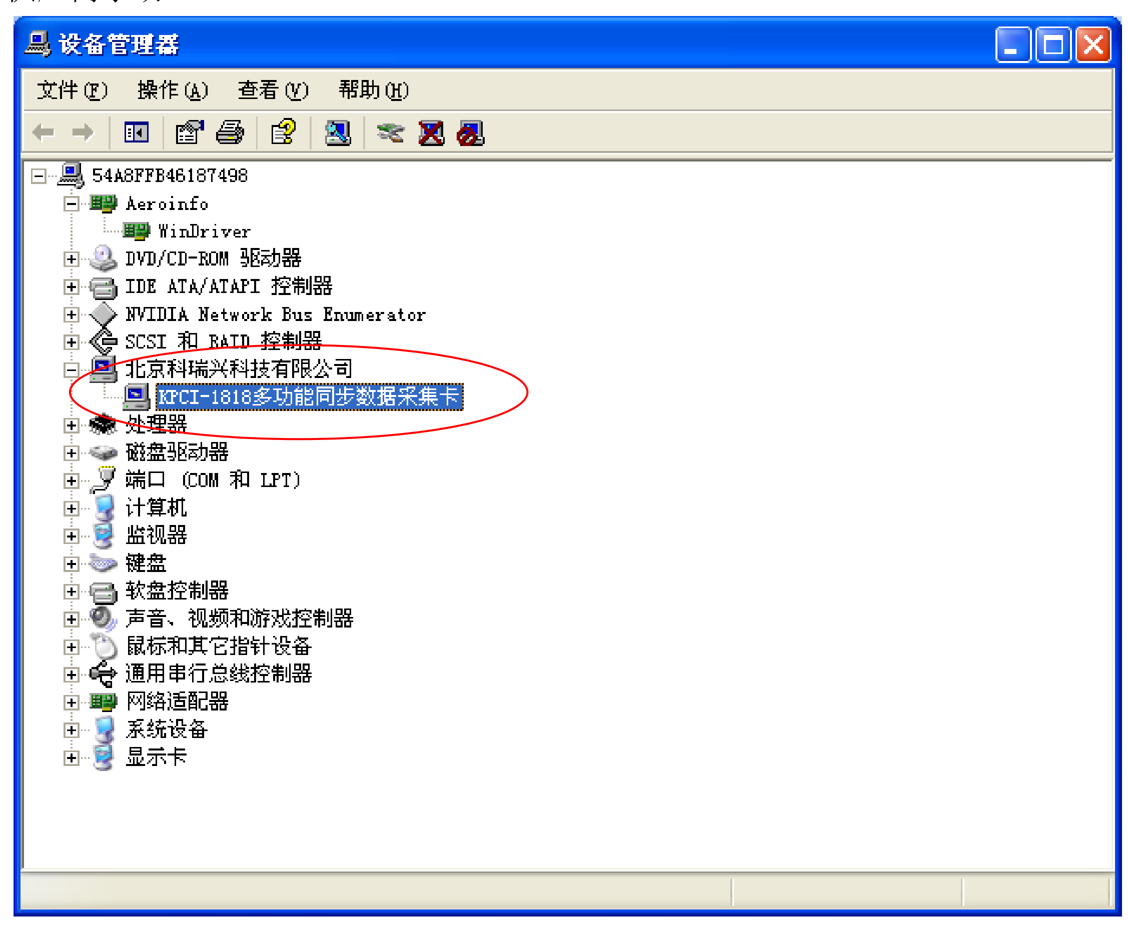


第六步 [找到新的硬件向导]，此步骤可能会出现 Windows 安装驱动程序的进度状态窗口，用户稍等片刻，然后出现[完成找到新硬件向导]对话框，单击 [完成]按钮。



二、安装结果验证

进入 Windows2000 [控制面板] 窗口，双击 [系统] 图标，弹出 [系统 特性] 对话框，在对话框中单击 [硬件] 标签页，然后单击 [设备管理器]按钮，进入 [设备管理器] 窗口，在 [本地计算机] 列表中检查是否有“北京科瑞兴业科技有限公司”显示，如下图所示。若有，表示 KPCI-1818 卡的驱动程序已成功安装，否则，说明您的安装过程出现了问题，请试着再安装，或向硬件供应商求助。



如果在 KPCI-1818 设备上出现黄色感叹号，可运行驱动程序文件夹中的 SETUP.bat，或把 KPCI1818S.SYS 文件拷贝到 windows\system32\drivers 目录下，重新扫描硬件。

三、PCI 设备软件测试系统的介绍

当您正确完成了第一或二节中的工作后，您便可以到光盘中“PCI 总线测控板卡\KPCI-1818 8 通道同步数据采集卡\编程示例” 文件夹，运行程序，进行板卡测试。当用户自己编制应用程序时，可以参照光盘中“PCI 总线测控板卡\ KPCI-18188 通道同步数据采集卡\编程示例” 文件夹中的 VB 和 VC 示例程序。

第四章 函数模块调用说明及编程实例

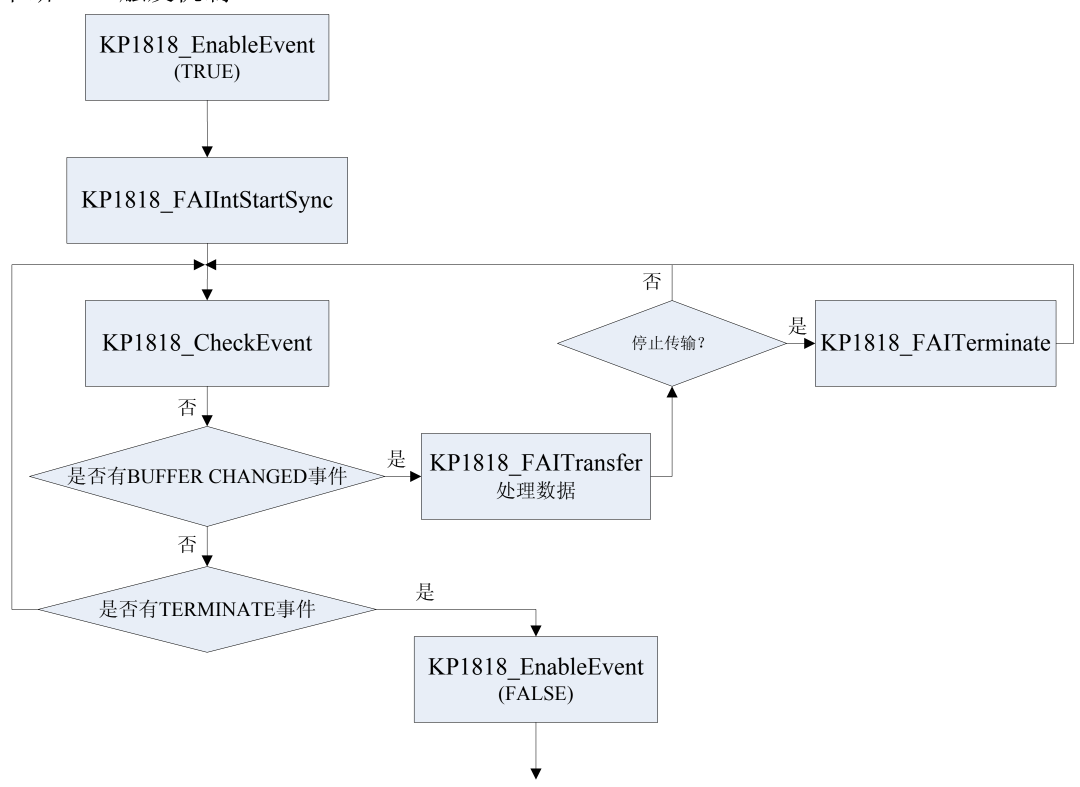
光盘中提供了在 VB，VC 和 WIN32 下开发应用的示例程序，给出了 KPCI-1818 卡的模拟量输入各种触发方式、相关参数的设置过程和数据读取方法，开关量输入输出的使用方法。用户可以参照相应程序段根据实际需要利用函数库中提供的函数设计自己的软件，而对数据采集过程中不处理其他任务的用户，也可以直接利用示例程序完成数据。初次使用动态链接库的用户还可以在程序中找到动态链接库的调用方法。为方便用户分析，示例程序以工程的形式提供了所有的资源和源代码。用户在编译时需要注意重新指定动态连接库的路径，或把动态连接库拷到指定的文件夹中。

一、函数说明：

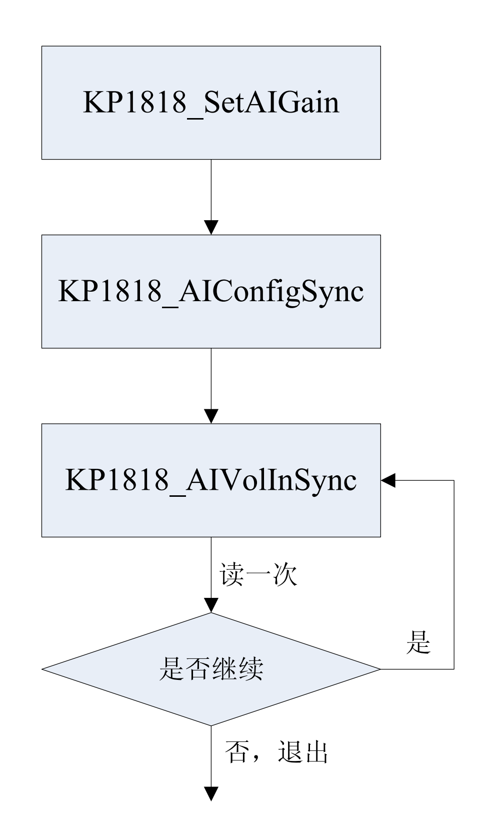
KPCI-1818 函数表	
函数名	函数功能
KP1818_DeviceOpen	打开要操作的设备
KP1818_SelectDevice	选择要操作设备的设备号
KP1818_DeviceClose	关闭指定的设备
KP1818_DeviceGetFeatures	获取设备特征
KP1818_DeviceConfig	配置和校验设备
KP1818_GetFifoSize	获取设备 FIFO 大小
KP1818_ReadPortByte	从指定I/O端口读一个字节的数据
KP1818_WritePortByte	写一个字节的数据的到指定的 I/O 端口
KP1818_ReadPortWord	从指定 I/O 端口读一个字的数据
KP1818_WritePortWord	写一个字的数据到指定的 I/O 端口
KP1818_AIGetConfig	获取指定的单个模拟量输入通道配置
KP1818_SetAIGain	设置 A/D 通道的增益
KP1818_AIConfigSync	配置指定的同步模拟量输入通道
KP1818_AIVolInSync	读取同步模拟量输入通道的电压值
KP1818_CheckEvent	检查是否有设定的事件发生
KP1818_FAIIntStartSync	启动同步模拟量采集
KP1818_FAITransfer	把采集到的数据传输到指定的缓存
KP1818_FAICheck	检查模拟量输入的状态
KP1818_FAITerminate	停止当前模拟量输入
KP1818_EnableEvent	启动或停止事件机制
KP1818_ClearOverrun	清除 Overrun 标志
KP1818_DioReadBit	读取指定通道的输入开关量
KP1818_DioWriteBit	写指定通道的输出开关量
KP1818_DioReadWord	读取某几个通道的输入开关量
KP1818_DioWriteWord	写入某几个通道的输出开关量
KP1818_DioGetCurrentDOWord	回读输出的开关量
KP1818_GetErrorMessage	根据错误代码返回错误信息
KP1818_GetAddress	该函数用于 VB，用于返回数据地址

二、函数调用流程：

1、中断FIFO触发机制



2、软件触发机制



三、函数说明：

1、设备操作

a. KP1818_DeviceOpen

Visual C++:

LRESULT KP1818_DeviceOpen (ULONG ulDeviceNum, PVOID *lpDev);

Visual Basic:

Declare Function KP1818_DeviceOpen Lib "KPCI1818.dll" (ByVal ulDeviceNum As Long,
lpDev As Long) As Long

功能：该函数打开要操作的设备。

参数：ulDeviceNum：是 Device Number（拨码开关）。

lpDev：是返回的设备指针

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

b. KP1818_SelectDevice

Visual C++:

LRESULT KP1818_SelectDevice(HWND hCaller, PULONG ulDevNum);

Visual Basic:

Declare Function KP1818_SelectDevice Lib "KPCI1818.dll" (ByVal hCaller As Long,
ulDevNum As Long) As Long

功能：该函数选择要操作设备的设备号。

参数：hCaller：调用者的窗口句柄；

ulDevNum：是返回的设备号（拨码开关）。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

c. KP1818_DeviceClose

Visual C++:

LRESULT KP1818_DeviceClose (PVOID lpDev)

Visual Basic:

Declare Function KP1818_DeviceCloseLib "KPCI1818.dll"(ByVal lpDev As Long) As Long

功能：该函数关闭设备。

参数：lpDev：设备指针。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

d. KP1818_DeviceGetFeatures

Visual C++:

LRESULT KP1818_DeviceGetFeatures(PVOID lpDev, DEVFEATURES far * DevFeature)

Visual Basic:

Declare Function KP1818_DeviceGetFeatures Lib "KPCI1818.dll" (ByVal lpDev As Long,
DevFeature As DEVFEATURES) As Long

功能：该函数获取设备特征。

参数：lpDev：设备指针；
DevFeature：是 DEVFEATURES 结构的指针，该结构包含了设备特征参数，结构定义请参照数据结构说明。
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

e. KP1818_DeviceConfig

Visual C++:

```
LRESULT KP1818_DeviceConfig (PVOID lpDev, HWND   hCaller)
```

Visual Basic:

```
Declare Function KP1818_DeviceConfig Lib "KPCI1818.dll" (ByVal lpDev As Long,
                                                         ByVal hCaller As Long) As Long
```

功能：该函数配置和校验设备。
参数：lpDev：设备指针；
hCaller：是调用该函数程序的窗口句柄。
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

f. KP1818_GetFifoSize

Visual C++:

```
LRESULT KP1818_GetFifoSize (PVOID lpDev, PULONG lSize);
```

Visual Basic:

```
Declare Function KP1818_GetFifoSize Lib "KPCI1818.dll" (ByVal lpDev As Long,
                                                        lSize As Long) As Long
```

功能：该函数获取设备 FIFO 大小。
参数：lpDev：设备指针；
lSize：是 long 的指针，该参数返回 FIFO 的大小。
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

2、I/O 端口操作

a. KP1818_ReadPortByte

Visual C++:

```
LRESULT KP1818_ReadPortByte (PVOID lpDev, USHORT port, USHORT * ByteData );
```

Visual Basic:

```
Declare Function KP1818_ReadPortByte Lib "KPCI1818.dll" (ByVal lpDev As Long,
                                         ByVal port As Integer, ByteData As Integer) As Long
```

功能：该函数读一个字节的 I/O 端口数据
参数：lpDev：设备指针；
port：是端口地址；
ByteData：从端口读取的字节数据。
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1818_WritePortByte

Visual C++:

LRESULT KP1818_WritePortByte(PVOID lpDev, USHORT port, USHORT ByteData);

Visual Basic:

Declare Function KP1818_WritePortByte Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal port As Integer, ByVal ByteData As Integer) As Long

功能：该函数写一个字节数据到 I/O 端口

参数：lpDev：设备指针；
port：是端口地址；
ByteData：写到端口的字节数据。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

c. KP1818_ReadPortWord

Visual C++:

LRESULT KP1818_ReadPortWord(PVOID lpDev, USHORT port, USHORT * WordData);

Visual Basic:

Declare Function KP1818_ReadPortWord Lib "KPCI1818.dll" (PVOID lpDev,
USHORT port, USHORT WordData) As Long

功能：该函数从 I/O 端口读一个字的数据

参数：lpDev：设备指针；
port：是端口地址；
WordData：从端口读取的字数据。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

d. KP1818_WritePortWord

Visual C++:

LRESULT KP1818_WritePortWord(PVOID lpDev, USHORT port, USHORT WordData);

Visual Basic:

Declare Function KP1818_WritePortWord Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal port As Integer, ByVal WordData As Integer) As Long

功能：该函数写一个字数据到 I/O 端口

参数：lpDev：设备指针；
port：是端口地址；
WordData：是写到端口的字数据。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

3、模拟量输入操作

a. KP1818_AIGetConfig

Visual C++:

LRESULT KP1818_AIGetConfig (PVOID lpDev,   LPDEVCONFIG_AI buffer);

Visual Basic:

Declare Function KP1818_AIGetConfig Lib "KPCI1818.dll" (ByVal lpDev As Long,
buffer As DEVCONFIG_AI) As Long

功能：该函数获取指定的单个模拟量输入通道的配置

参数：lpDev：设备指针；

buffer：是 DEVCONFIG_AI 结构的指针，该结构包含要模拟量输入通道配置的参数，结构定义请参照数据结构说明。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

b. KP1818_SetAIGain

Visual C++:

LRESULT KP1818_SetAIGain (PVOID lpDev, USHORT gain);

Visual Basic:

Declare Function KP1818_SetAIGain Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal gain As Integer) As Long

功能：该函数配置指定模拟量输入通道的增益。

参数：lpDev：设备指针；

gain：是采样增益的代码。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

c. KP1818_AIConfigSync

Visual C++:

LRESULT KP1818_AIConfigSync (PVOID lpDev, USHORT SyncNum, USHORT adBits);

Visual Basic:

Declare Function KP1818_AIConfigSync Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal SyncNum As Integer, ByVal adBits As Integer) As Long

功能：该函数配置指定的单个模拟量输入通道。

参数：lpDev：设备指针；

SyncNum：同步通道数；

adBits：采样位数；

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

d. KP1818_AIVolInSync

Visual C++:

LRESULT KP1818_AIVolInSync (PVOID lpDev, USHORT TrigMode, FLOAT far * VoltageArray);

Visual Basic:

Declare Function KP1818_AIVolInSync Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal TrigMode As Integer, ByVal VoltageArray As Long) As Long

功能：该函数读取模拟量输入的电压值

参数：lpDev：设备指针；

TrigMode：触发模式；

VoltageArray：已转换成伏特值的形式的输入的电压；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

4、高速数据采集操作

a. KP1818_CheckEvent

Visual C++:

LRESULT KP1818_CheckEvent (PVOID lpDev, USHORT far *EventType, DWORD Milliseconds);

Visual Basic:

Declare Function KP1818_CheckEvent Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByRef EventType As Integer, ByVal Milliseconds As Long) As Long

功能：该函数检查是否有设定的事件发生

参数：lpDev：设备指针；
EventType：希望检测的事件类型；
Milliseconds：事件超时的毫秒数；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1818_FAIIntStartSync

Visual C++:

LRESULT KP1818_FAIIntStartSync (PVOID lpDev, USHORT TrigSrc, DWORD SampleRate,
USHORT SyncNum, USHORT far * buffer, ULONG count,
USHORT cyclic, USHORT IntrCount, USHORT usMode, USHORT adBits);

Visual Basic:

Declare Function KP1818_FAIIntStartSync Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal TrigSrc As Integer, ByVal SampleRate As Long, ByVal SyncNum As Integer,
ByVal buffer As Long, ByVal Count As Long, ByVal Cyclic As Integer,
ByVal IntrCount As Integer, ByVal usMode As Integer, ByVal adBits As Integer) As Long

功能：该函数启动同步模拟量采集

参数：lpDev：设备指针；
TrigSrc：触发源: 1 位外部触发源, 0 位内部触发源；
SampleRate：采样频率；
SyncNum：同步采样通道数；
buffer：用户分配的、用于存储数据的缓存；
count：采样的次数；
cyclic：是否循环模式: 1 位循环模式, 0 位非循环模式；
IntrCount：采样多少次一中断，（只有 FIFO 和中断两种采样模式，中断为 1，FIFO 则为半满 FIFO 大小）；
usMode：同步卡采样模式：0 为无模式，1 为主模式，2 为从模式；
adBits：采样的位数：0 为 12 位，1 为 16 位；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

c. KP1818_FAITransfer

Visual C++:

LRESULT KP1818_FAITransfer (PVOID lpDev, USHORT ActiveBuf, LPVOID DataBuffer,
USHORT DataType, ULONG start, ULONG count, USHORT far *overrun);

Visual Basic:

Declare Function KP1818_FAITransfer Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal ActiveBuf As Integer, ByVal DataBuffer As Long,
ByVal DataType As Integer, ByVal start As Long,
ByVal Count As Long, overrun As Long) As Long

功能：该函数把采集到的数据传输到指定的缓存

参数：lpDev：设备指针；

ActiveBuf：Buffer 的类型，本设备总是为 0；

DataBuffer：当数据类型为原始数据，则缓冲长度应该不小于 2*count。当数据类型为电压值时，则缓冲长度应该不小于 4*count；

DataType：数据类型，0 为原始数据，1 为电压值（浮点数）；

Start：从源缓冲复制到用户缓冲的起始点；

Count：从源缓冲复制到用户缓冲的采样个数；

Overrun：overrun 状态：0- 无 overrun 发生，1- Overrun 发生了

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

d. KP1818_FAICheck

Visual C++:

LRESULT KP1818_FAICheck (PVOID lpDev, USHORT far * ActiveBuf, USHORT far * stopped,
ULONG far * retrieved, USHORT far * overrun, USHORT far * HalfReady);

Visual Basic:

Declare Function KP1818_FAICheck Lib "KPCI1818.dll" (ByVal lpDev As Long,
ActiveBuf As Long, stopped As Long,
retrieved As Long, overrun As Long, HalfReady As Long) As Long

功能：该函数检查模拟量输入的状态

参数：lpDev：设备指针；

ActiveBuf：Buffer 的类型，本设备总是为 0；

Stopped：说明操作是否完成，0 代表未完成，1 代表完成 。

Retrieved：A/D 转换的次数；

Overrun：是否有 overrun 发生， 0 无 overrun， 1 有 overrun；

HalfReady：说明 buffer 状态， 0 为数据没准备好； 1 为第一次半满 buffer 准备. 2 为第二次半满 buffer 准备。

返回值：如果执行成功，则返回 0；

如果未成功，则返回错误代码。

e. KP1818_FAITerminate

Visual C++:

LRESULT KP1818_FAITerminate (PVOID lpDev);

Visual Basic:

Declare Function KP1818_FAITerminate Lib "KPCI1818.dll" (ByVal lpDev As Long) As Long

功能：该函数停止当前模拟量输入

参数：lpDev：设备指针

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

f. KP1818_EnableEvent

Visual C++:

LRESULT KP1818_EnableEvent (PVOID lpDev, USHORT EventType,
USHORT Enabled, USHORT Count);

Visual Basic:

Declare Function KP1818_EnableEvent Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal EventType As Integer, ByVal Enabled As Integer,
ByVal Count As Integer) As Long

功能：该函数启动或停止事件机制

参数：lpDev：设备指针；
EventType：希望触发的事件类型；
Enabled：使能或取消触发的事件类型。1 为使能；0 为取消
Count：希望事件触发的次数；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

g. KP1818_ClearOverrun

Visual C++:

LRESULT KP1818_ClearOverrun(PVOID lpDev);

Visual Basic:

Declare Function KP1818_ClearOverrun Lib "KPCI1818.dll" (ByVal lpDev As Long) As Long

功能：该函数清除 Overrun 标志

参数：lpDev：设备指针；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

5、开关量输入/输出操作

a. KP1818_DioReadBit

Visual C++:

LRESULT KP1818_DioReadBit(PVOID lpDev, USHORT bit, USHORT far * state);

Visual Basic:

Declare Function KP1818_DioReadBit Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal bit As Integer, state As Long) As Long

功能：该函数读一个字节的 I/O 端口数据
参数：lpDev：设备指针；
bit：要读取的位；
state：读回的状态；
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1818_DioWriteBit

Visual C++:

LRESULT KP1818_DioWriteBit(PVOID lpDev, USHORT bit, USHORT state);

Visual Basic:

Declare Function KP1818_DioWriteBit Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal bit As Integer, ByVal state As Integer) As Long

功能：该函数读一个字节的 I/O 端口数据
参数：lpDev：设备指针；
bit：要写入的位；
state：要写入的位的状态；
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

c. KP1818_DioReadWord

Visual C++:

LRESULT KP1818_DioReadWord(PVOID lpDev, USHORT * ValidChannelMask,
USHORT * value);

Visual Basic:

Declare Function KP1818_ DioReadWord Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByRef ValidChannelMask As Integer, ByRef value As Integer) As Long

功能：该函数从 I/O 端口读一个字的数据
参数：lpDev：设备指针；
ValidChannelMask：DI 通道掩码；
Value：读回的 DI 值；
返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

d. KP1818_DioWriteWord

Visual C++:

LRESULT KP1818_DioWriteWord (PVOID lpDev, USHORT mask, USHORT state);

Visual Basic:

Declare Function KP1818_ DioWriteWord Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByVal mask As Integer, ByVal state As Integer) As Long

功能：该函数写一个字节数据到 I/O 端口
参数：lpDev：设备指针；
mask：DO 通道的掩码；
state：写入的 DO 值；

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

e. KP1818_DioGetCurrentDOWord

Visual C++:

LRESULT KP1818_DioGetCurrentDOWord (PVOID lpDev, USHORT far * value);

Visual Basic:

Declare Function KP1818_DioGetCurrentDOWord Lib "KPCI1818.dll" (ByVal lpDev As Long,
ByRef value As Integer) As Long

功能：该函数回读输出的开关量。

参数：lpDev：设备指针；
value：是所读出的输出开关量的值。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

6、其他操作

a. KP1818_GetErrorMessage

Visual C++:

LRESULT KP1818_GetErrorMessage(LRESULT lError, TCHAR* lpMsg);

Visual Basic:

Declare Function KP1818_GetErrorMessage Lib "KPCI1818.dll" (ByVal lError As Long,
ByVal lpszszErrMsg As String) As Long

功能：该函数根据错误代码返回错误信息

参数：lError：错误代码，lpMsg：指向错误信息字符串的指针。

返回值：如果执行成功，则返回 0；
如果未成功，则返回错误代码。

b. KP1818_GetAddress

Declare Function KP1818_GetAddress Lib "KPCI1818.dll" (lpVoid As Any) As Long

功能：该函数用于 VB，返回数据地址

参数：lpVoid：可以是任意的类型，需要返回其地址

返回值：返回该数据的地址。

四、数据结构说明

1、设备操作

a）DEVFEATURES

```
typedef struct tagDEVFEATURES
{
    USHORT    usMaxAIDiffChl;
    USHORT    usMaxAISiglChl;
    USHORT    usMaxTimerChl;
    USHORT    usMaxAlarmChl;
```

```
    USHORT    usNumADBit;
    USHORT    usNumADByte;
    USHORT    usNumGain ;
    GAINLIST glGainList[32];
    DWORD     dwPermutation[4];
} DEVFEATURES, FAR * LPDEVFEATURES;
```

Member Description:

名称	方向	类型	描述
usMaxAIDiffChl	Output	USHORT	差分模拟量输入通道的最大个数
usMaxAISiglChl	Output	USHORT	单端模拟量输入通道的最大个数
usMaxTimerChl	Output	USHORT	计数器/时钟的最大个数
usMaxAlarmChl	Output	USHORT	报警通道的最大个数
usNumADBit	Output	USHORT	A/D 转换的位数
usNumADByte	Output	USHORT	A/D 转换所需字节个数
usNumGain	Output	USHORT	输入范围的最大个数
glGainList	Output	**GAINLIST** 数组	所有支持输入范围
dwPermutation	Output	DWORD	设备所支持功能

Note：

dwPermutation 的定义：

Bit 0: Software AI
Bit 1: DMA AI
Bit 2: Interrupt AI
Bit 3: Condition AI
Bit 4: Software AO
Bit 5: DMA AO
Bit 6: Interrupt AO
Bit 7: Condition AO
Bit 8: Software DI
Bit 9: DMA DI
Bit 10: Interrupt DI
Bit 11: Condition DI
Bit 12: Software DO
Bit 13: DMA DO
Bit 14: Interrupt DO
Bit 15: Condition DO
Bit 16: High Gain
Bit 17: Auto Channel Scan
Bit 18: Pacer Trigger
Bit 19: External Trigger
Bit 20: Down Counter
Bit 21: Dual DMA
Bit 22: Monitoring
Bit 23: Qcounter

b) GAINLIST

```
typedef struct tagGAINLIST
{
    USHORT usGainCde;
    FLOAT fMaxGainVal;
    FLOAT fMinGainVal;
    CHAR szGainStr[16];
} GAINLIST;
```

Member Description:

名称	方向	类型	描述
usGainCde	Output	USHORT	该模拟量输入范围的代码
fMaxGainVal	Output	float	该模拟量输入范围的最大值
fMinGainVal	Output	float	该模拟量输入范围的最小值
szGainStr	Output	Char 数组	该模拟量输入范围的字符描述

c) DEVCONFIG_AI

```
typedef struct tagDEVCONFIG_AI
{
    DWORD     ulChanConfig;
    USHORT    usGainCtrMode;
    USHORT    usPolarity;
    USHORT    usDasGain;
    USHORT    usCjcChannel;
} DEVCONFIG_AI, FAR * LPDEVCONFIG_AI;
```

Member Description:

名称	方向	类型	描述
ulChanConfig	Output	DWORD	AI 控制位：0 代表单端，1 代表差分
usGainCtrMode	Output	USHORT	AI 量程控制：1 代表跳线控制，0 代表程序控制
usPolarity	Output	USHORT	AI 极性：0 为双极性，1 为单极性
usDasGain	Output	USHORT	AI 量程代码（usGainCtrMode=1 不可用）
usCjcChannel	Output	USHORT	保留

第五章　KPCI-1818保修和注意事项

一、应用注意事项 ：

在公司售出的产品包装中，用户将会在资料光盘中找到本卡的说明书。

在使用KPCI-1818卡时，应注意以下问题：

① KPCI-1818卡正面的IC芯片不要用手去摸，防止芯片受到静电的损害。

② 在使用KPCI-1818卡时，可通过K-801E等信号调理端子板与现场信号连接。

③ 用户务必注意电源的开关顺序，使用时要求先开主机电源，后开信号源的电源；先关信号源的电源，后关主机电源。

二、A/D的校准：

KPCI-1818是8通道同步数据采集，8个通道是互相独立的，，每个通道都有零点和满度调整电位器，需要分别调整，KPCI-1818卡在出厂前已经按照±5V量程调整好，如果用户想重新调整，可以按照下面发发调整，下面仅以第一通道为例，介绍零点和满度的调整方法，其他通道类似。

（一） **单极性输入信号的校准**

1） 将卡上的JP1短路套连接2、3脚。

2） ① 零点调整

KPCI-1818的Ain1+和Ain1-之间接0伏，在Windows下运行KPCI-1818测试程序,选择1通道，选择单通道数据采集方式，采集开始后，调整电位器VR2，使显示值为 0。

② 满度调整

KPCI-1818的Ain1+和Ain1-之间接正满度电压+9998 mV，调整电位器VR1，使显示值为 +9998mV。

重复以上步骤，直到满足要求为止。

（二） **双极性输出的校准**

1） 将卡上的JP1短路套连接1、2脚。

① 零点调整

KPCI-1818的Ain1+和Ain1-之间接0伏，在Windows下运行KPCI-1818测试程序，选择1通道，选择单通道数据采集方式，采集开始后，调整电位器VR2，使显示值为 0。

② 满度调整

KPCI-1818的Ain1+和Ain1-之间接正满度电压+4998 mV，调整电位器VR1，使显示值为 +4998mV。

将 Ain1+和Ain1-之间接负满度电压-4998mV，看显示值是否为 -4998mV。

重复以上步骤，直到满足要求为止。

三、保修：

KPCI-1818自出厂之日起，两年内凡用户遵守贮存，运输和使用要求，而产品质量低于技术指标的，凭保修卡免费维修。因违反操作规定和要求而造成损坏的，需交纳器件维修费。

四、产品配套清单：

4.1 KPCI-1818 8通道同步数据采集卡壹块。
4.2 北京科瑞兴业公司产品光盘壹张。
4.3 37芯D型插头壹套。